미정보 환경 하에서 신경회로망 힘추종 로봇 제어 기술의 실험적 연구

Experimental Studies on Neural Network Force Tracking Control Technique for Robot under Unknown Environment

정 슬, 임 선 빈
(Seul Jung and Sun Bin Yim)

Abstract : In this paper, neural network force tracking control is proposed. The conventional impedance function is reformulated to have direct force tracking capability. Neural network is used to compensate for all the uncertainties such as unknown robot dynamics, unknown environment stiffness, and unknown environment position. On line training signal of force error for neural network is formulated. A large x-y table is built as a test-bed and neural network learning algorithm is implemented on a DSP board mounted in a PC. Experimental studies of force tracking on unknown environment for x-y table robot are presented to confirm the performance of the proposed technique.

Keywords : impedance force control, robot manipulator, neural network, training signal, uncertainties

I. 서론

최근의 산업로봇은 단순히 물건을 옮기는 작업으로부터 엔진 주물의 버(burr)를 제거한다던가, 자동차의 후드(hood)의 광택작업을 한다던가, 더 나아가 두 대의 로봇이 서로 협조하여 작업하는 어려운 작업을 수행하도록 요구되어지고 있다. 이러한 작업은 로봇이 대상과 관계를 이루며 작업을 해야 하는 어려운 점이 있다. 로봇이 어떤 대상에 접촉하여 작업을 할 때에는 접촉을 유지하기 위하여 그 대상에 일정한 힘을 가한 상태에서 작업해야 하므로 적당한 힘을 제어하는 기술, 즉 힘제어 기술이 필요하다. 예컨대, 두 대의 로봇이 한 물건을 잡고 서로 협조하며 작업한다는 것은 위치제어 만으로는 어려우며 반드시 힘제어를 수반하여야 한다. 이처럼 힘제어는 고 난이도의 로봇 제어 방식으로 산업 공정에서 뿐만 아니라 최근 많은 인기를 얻고 있는 Haptic interface에도 반드시 필요한 제어 기술이다.

힘제어는 위치제어가 기반이 되므로 먼저 위치제어가 잘 되어야 한다. 또한 힘제어는 대상과의 상호작용에 의해 힘이 결정되는 임피던스 힘제어 방식과 힘제어와 위치제어를 나누어 직접 원하는 힘을 제어하는 하이브리드 힘제어 방식으로 크게 둘로 나뉜다[1][2].

이전의 논문에서는 적응제어 방식을 적용하여 미정보 환경 하에서 원하는 힘을 추종하는 힘제어 방식을 제안하였다[3]. 로봇의 동적 불확실성은 시간 지연 제어 방식(Time-delayed control)을 사용하여 보상하였고 미정보 환경의 불확실성, 예를 들면 대상의 위치와 강성도의 정확한 정보의 부재에서 오는 불확실성은 적응제어 방식을 사용하여 보상하였다.

본 논문에서는 동적 모델의 불확실성과 대상으로부터 초래되는 불확실성 등 모든 것을 하나의 신경회로망으로 보상하는 방식을 제안하고자 한다. 신경회로망은 학습능력과 적응능력, 비선형 맵핑능력 등 복잡하고 어려운 비선형 시스템의 문제점을 보완하는 도구로 많이 사용되어져 왔다. 제어분야에서는 비선형 제어기로 많이 사용되어져 왔으며, 최근에는 빠른 연산의 하드웨어의 지원으로 실시간 제어가 가능해졌다. 최근에는 신경회로망을 이용한 로봇제어의 분야에 대한 관심도가 증가하고 있으며 실험적인 연구 결과가 많이 도출되고 있는 실정이다. 신경회로망이 산업 로봇이나 유연 로봇의 비선형성을 보상하여 위치 추종을 잘하도록 한다[4][5]. 또한 이동로봇의 제어나 해저로봇 분야에도 신경망을 적용하고 있다[6][7]. 하지만 신경망을 이용한 힘제어 기술 분야는 아직 미개척분야이고 실험적 결과를 찾아보기란 어렵다.

따라서, 본 논문에서는 신경회로망을 이용한 로봇의 위치제어에 대한 기존의 연구를 연장하여 신경회로망을 힘제어에 적용하여 실험적으로 검증하고자 한다[8][9]. 신경회로망의 많은 연산을 실시간으로 계산하기 위해서 DSP 보드를 사용하였으며 대상의 불확실성을 실험하기 위해 강성도가 다른 나무와 철판을 볼록하게 만들어 로봇이 추종하도록 하였다.

II. 직교공간에서의 로봇 동력학

관절이 n인 로봇의 동적 방정식은 다음과 같다.

$$D(q)\ddot{q}+H(q,\dot{q})=\tau-\tau_e \tag{1}$$

$D(q)$는 n x n 관성 행렬을 나타내고, $H(q,\dot{q})$는 n x 1 코리올리스힘, 원심력과 중력 벡터를 나타내고, q, $\dot{q}$, $\ddot{q}$는 각각 n x 1 관절의 각, 각속도, 각가속도 벡터를 나타내고, τ는 n x 1 토크 벡터, 그리고 τ_e는 n x 1 외부의 토크벡터를 나타낸다.

힘제어는 카테시안 공간에서 이루어지므로 관절 공간의

접수일자 : 2001. 7. 12., 수정완료 : 2002. 1. 10.
정 슬, 임선빈 : 충남대학교 메카트로닉스공학과(jungs@cnu.ac.kr/s_isbn@cnu.ac.kr)

동력학식을 카테시안 공간으로 변환해야 한다. 조인트 속도를 직교공간의 속도로 변환하기 위해서는 자코비안 행렬을 통해 다음과 같이 표현된다.

$$\dot{q}=J^{-1}\dot{X} \tag{2}$$

여기서 X는 n x 1 직교좌표의 위치 및 오리엔테이션 벡터이고 J는 n x n 자코비안 행렬이다.

(2)를 미분하면 각가속도를 얻는다.

$$\ddot{q}=J^{-1}(\ddot{X}-\dot{J}\dot{q}) \tag{3}$$

(2)와 (3)을 (1)에 대입하고 정리하면 직교 공간에서의 로봇 동적 모형을 다음과 같이 구할 수 있다.

$$D^*(X)\ddot{X}+H^*(X,\dot{X})=F-F_e \tag{4}$$

$D^*=J^{-T}DJ^{-1}$는 직교 공간에서의 관성 행렬, $H^*=J^{-T}H-D^*\dot{J}J^{-1}\dot{X}$는 직교공간에서의 코리올리스힘, 원심력과 중력을 나타내고, F는 실제 힘, F_e는 외부로부터의 힘을 나타낸다. (4)는 비선형이고 각 관절이 서로 연관되어 있기 때문에 선형화하고 분리시켜 제어한다.

III. 임피던스 힘제어 방식 및 동적 불확실성

임피던스 힘제어 방식은 말 그 자체에서 알 수 있듯이 전기회로에서 입력과 출력의 임피던스 관계를 나타낸다. 이는 스프링 댐퍼 시스템과 연결지어 힘과 위치와의 관계를 임피던스 관계식으로 나타낼 수 있게 한다[1]. 임피던스 방식은 임피던스 변수들의 값을 조절하므로 힘을 조절하게 되는 간접제어 방식을 나타낸다.

먼저 힘제어를 적용하기에 앞서 위치제어를 보상하여만 힘제어를 잘 할 수 있게 된다. 카테시안 공간에서 로봇 모델을 기반으로 한 제어법칙은 다음과 같다.

$$F=\hat{D}^*V+\hat{H}^*+F_e \tag{5}$$

여기에서 $\hat{D}^*$와 $\hat{H}^*$는 각각 카테시안 공간의 모델 D^*와 H^*의 평가치(estimation)이다.

여기서 제어입력 V는 다음과 같이 임피던스 관계식으로부터 설정한다.

$$\ddot{X}\cong V=\ddot{X}_r+M^{-1}(B\dot{E}+KE-F_e) \tag{6}$$

여기서, $E=X_r-X$, X_r은 주어진 경로이고, X는 실제 경로이다. 또한 M은 관성 이득 행렬, B는 댐핑이득 행렬, K는 강성이득 행렬이고 F_e는 외부로부터의 힘으로 힘센서에 의해 측정된다. 실제 로봇 동적 방정식(4)에 로봇의 모델 방정식(5)(6)을 대입하면 다음과 같은 오차 방정식이 유도된다.

$$M\ddot{E}+B\dot{E}+KE=F_e+\Delta_M \tag{7}$$

여기서, $\Delta_M=\hat{D}^{*-1}[\Delta D^*\ddot{X}+\Delta H^*]$로 로봇의 동적 불확실성을 나타내는데, $\Delta D^*=D^*-\hat{D}^*$, $\Delta H^*=H^*-\hat{H}^*$이다.

결과적으로, 임피던스 제어 방법을 사용하여 힘제어를 정확하게 하려면 우선적으로 로봇의 위치제어가 보장되어야 한다. 임피던스 제어 방식은 로봇의 모형을 근거로 하는 토크 계산 방식을 기본으로 하기 때문에 (7)의 로봇 모형의 불일치에서 오는 불확실성이 보상되지 않으면 힘제어를 잘 할 수 없게 된다.

(7)에서 보여진 것처럼 힘제어 방식들의 수행 능력은 로봇 동적 모형의 정확도와 대상의 불확실성들에 따라 좌우된다. 실제로는 로봇 동적 모형을 정확하게 구할 수가 없기 때문에 모형의 오차가 생기고 마찰력이나 백래쉬(backlash) 등과 같은 비선형 요소들과 알지 못하는 디버링 도구의 영향이 존재하기 때문에 정확한 위치 제어 수행을 저해한다. 이러한 위치 제어의 오차는 결국 힘제어의 오차를 초래하기 때문에 그에 대한 보상이 필요하다[4].

(7)에서 불확실성 Δ_M가 보상되어지면, 힘이 제어되는 방향에서의 임피던스 함수는 다음과 같이 이상적인 형태로 나타날 수 있다.

$$F_e=M\ddot{E}+B\dot{E}+KE \tag{8}$$

정상상태(steady state)에서 $F_e=KE=K(X_r-X)$이므로 주어진 경로 X_r과 강성이득 K를 정함에 따라서 힘의 크기가 결정됨을 알 수 있다. 주어진 경로 X_r은 원하는 힘 F_d, 대상의 강성도 K_e, 대상의 위치 X_e, 그리고 강성 이득 K 등의 값들을 정확하게 알아야 만 다음 식을 사용하여 미리 계산을 할 수 있다.

$$\begin{aligned} X_r&=X_e+\Delta X_r=X_e+K_{eff}^{-1}F_d,\\ K_{eff}^{-1}&=K^{-1}+K_e^{-1}\end{aligned} \tag{9}$$

F_d와 K는 사용자가 정할 수 있지만 K_e나 X_e는 사용자가 쉽게 정할 수 없다. 따라서, X_e의 값을 얻기 위한 시각 센서의 도움 없이 힘제어의 수행 능력을 향상시키기 위해서는 정확한 대상의 강성도 K_e의 정보와 대상의 위치 X_e의 정보가 필요하다. 이처럼 임피던스 제어에서는 임피던스 매개 변수를 대상에 맞게 선택하여 주어진 경로를 조절함으로써 원하는 힘을 간접적으로 조절하게 된다.

IV. 힘추종 제어 방식

임피던스 관계식(8)에서 추종하고자 하는 힘 F_d를 빼면 다음과 같은 새로운 임피던스 관계식을 얻게 된다.

$$F_e-F_d=M\ddot{E}+B\dot{E}+KE \tag{10}$$

정상상태에서는 다음과 같다.

$$F_e-F_d=K(X_r-X) \tag{11}$$

따라서 원하는 힘을 추종하기 위해서는 마찬가지로 강성과 이득값의 역할이 매우 큼을 알 수 있다. 그러므로 힘을 직접적으로 제어하기 위해서는 다음과 같이 임피던스 함수를 구성하여야 한다.

$$F_e - F_d = M\ddot{E} + B\dot{E} \tag{12}$$

$F_e = K_e(X - X_e)$이므로 대입하면 다음과 같다.

$$F_d = -M\ddot{E} - B\dot{E} - K_e(X - X_e) \tag{13}$$

새로운 오차 $\varepsilon = X_e - X$, X_e는 대상의 정확한 위치, 를 정의하고 (13)을 다시 정리하면 다음과 같다.

$$F_d = M\ddot{\varepsilon} + B\dot{\varepsilon} + K_e\varepsilon \tag{14}$$

정상상태에서는

$$F_e = F_d \tag{15}$$

를 만족한다.

하지만 일반적으로 대상의 위치를 정확하게 알지 못하므로 대상의 위치에 불확실성 ΔX_e을 포함하면 다음과 같다.

$$X_e' = X_e + \Delta X_e \tag{16}$$

(14)에 대입하고 정리하면 다음과 같다.

$$\begin{aligned} F_d &= M\ddot{\varepsilon}' + B\dot{\varepsilon}' + K_e\varepsilon' \\ &= M\ddot{\varepsilon} + B\dot{\varepsilon} + K_e\varepsilon + M\Delta\ddot{X}_e + B\Delta\dot{X}_e + K_e\Delta X_e \end{aligned} \tag{17}$$

정상상태에서 (17)은 다음과 같으므로 정확한 힘을 추종하지 못하게 된다.

$$F_d - F_e = K_e\Delta X_e \tag{18}$$

(17)의 로봇 동력학의 불확실성을 포함하면 (14)는 다음과 같이 된다.

$$F_d - F_e = M\ddot{\varepsilon}' + B\dot{\varepsilon}' + \Delta_M \tag{19}$$

결과적으로 힘추종은 어렵게 된다.

V. 신경망 제어 및 학습방식

앞에서 언급한 대상과 로봇 동력학으로부터의 불확실성을 보상하기 위해 신경회로망을 사용한다. 신경회로망은 비선형 학습 능력이 있으므로 비선형 시스템 제어에 매우 유용하게 사용할 수 있다[8].

신경회로망의 출력을 제어 입력에 더하면 다음과 같다.

$$\begin{aligned} V_p &= \ddot{X}_d + M^{-1}(B\dot{E} + KE) + \Phi_p \\ V_f &= \ddot{X}_e' + M^{-1}(B\dot{\varepsilon}' + F_d - F_e) + \Phi_f \end{aligned} \tag{20}$$

여기서, V_p는 위치제어 방향의 제어입력이고 V_f는 힘제어 방향의 제어입력이다.

(20)을 (5)에 대입하고 (4)와 등식으로 놓으면 다음과 같다.

$$\begin{aligned} &\hat{D}^*(\ddot{X}_e' + M^{-1}(B\dot{\varepsilon}' + F_d - F_e) + \Phi_f) + \hat{H}^* + F_e \\ &= D^*\ddot{X} + H^* + F_e \end{aligned} \tag{21}$$

(21)에 $\hat{D}^*\ddot{X}$를 더하고 빼준 뒤, 정리하면

$$\begin{aligned} &\hat{D}^*(\ddot{X}_e' - \ddot{X} + M^{-1}(B\dot{\varepsilon}' + F_d - F_e) + \Phi_f) \\ &= D^*\ddot{X} - \hat{D}^*\ddot{X} + H^* - \hat{H}^* \end{aligned} \tag{22}$$

(22)의 양변을 $\hat{D}^*$로 나누면 다음과 같다.

$$\begin{aligned} &\ddot{\varepsilon}' + M^{-1}(B\dot{\varepsilon}' + F_d - F_e) + \Phi_f \\ &= \hat{D}^{*-1}(D^*\ddot{X} - \hat{D}^*\ddot{X} + H^* - \hat{H}^*) \end{aligned} \tag{23}$$

(23)의 양변에 M을 곱하고 정리하면 그림1과 같은 제어블록이 되며 다음과 같은 식을 얻게 된다

$$M\ddot{\varepsilon}' + B\dot{\varepsilon}' + F_d - F_e = M\hat{D}^{*-1}(\Delta\hat{D}\ddot{X} + \Delta\hat{H}) - M\Phi_f \tag{24}$$

여기서, $\Delta\hat{D}^* = D^* - \hat{D}^*$, $\Delta H = H^* - \hat{H}^*$이다.

마찬가지로 하면 위치제어 방향의 식은 다음과 같이 얻게 된다.

$$M\ddot{E} + B\dot{E} + KE = M\hat{D}^{*-1}(\Delta\hat{D}\ddot{X} + \Delta\hat{H}) - M\Phi_p \tag{25}$$

여기서, $E = X_r - X$이다.

(24)에서 힘제어 방향의 신경회로망의 학습신호를 다음과 같이 정의한다.

$$\begin{aligned} e_f &= F_d - F_e \\ &= -M\ddot{\varepsilon}' - B\dot{\varepsilon}' + M\hat{D}^{*-1}(\Delta\hat{D}\ddot{X} + \Delta\hat{H}) - M\Phi_f \end{aligned} \tag{26}$$

수렴되었을 경우, 즉 $e_f \to 0, F_e \to F_d$일 때, 신경망의 출력은 다음과 같다.

$$\Phi_f \simeq -\ddot{\varepsilon}' - M^{-1}B\dot{\varepsilon}' + \hat{D}^{*-1}(\Delta\hat{D}\ddot{X} + \Delta\hat{H}) \tag{27}$$

마찬가지로 위치제어방향의 학습신호는 다음과 같이 정의한다.

$$e_p = M\ddot{\varepsilon}' + B\dot{\varepsilon}' + K\varepsilon \tag{28}$$

그러므로 힘추종 오차 e_f와 위치추종오차 e_p를 최소화하도

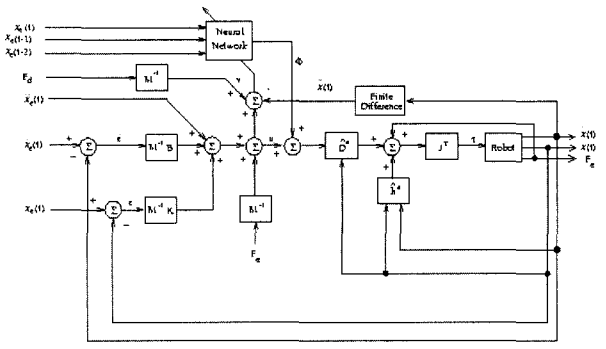


그림 1. 신경망 제어 블록 다이아 그램.
Fig. 1. Neural network control block diagram.

록 역전파 알고리즘을 설계하여 신경망을 학습시키면 모든 불확실성이 제거된다.

신경회로망을 학습시키기 위해 최소화하고자 하는 목적함수는 다음과 같다.

$$E=\frac{1}{2}e^{T}e \tag{29}$$

여기서, e는 e_p와 e_f를 포함하는 오차벡터이다.

(29)를 중치에 대해 미분하면 목적함수 E의 그래디언트를 다음과 같이 구할 수 있다.

$$\frac{\partial E}{\partial w}=\frac{\partial e^{T}}{\partial w}e \tag{30}$$

여기서, 그래디언트 $\frac{\partial e^{T}}{\partial w}$는 (24)와 (25)로부터 다음과 같이 구할 수 있다.

$$\frac{\partial e^{T}}{\partial w}=-M\frac{\partial \Phi^{T}}{\partial w} \tag{31}$$

(31)은 가중값들을 다음과 같이 새롭게 하기 위해 역전파 (back-propagation) 알고리즘에 사용된다.

$$\Delta w(t)=-\eta\frac{\partial E}{\partial w}+\alpha\Delta w(t-1) \tag{32}$$

$$w(t+1)=w(t)+\Delta w(t) \tag{33}$$

여기서 η값은 학습률이고, α는 운동량(momentum) 상수이다.

VI. 실험 환경

1. 전체 시스템 실험 환경

아래 그림 2는 x-y 테이블을 제어하기 위해 PC의 DSP 보드와 모터 드라이버 그리고 x-y 테이블의 관계를 나타낸다. PC 화면상의 GUI 프로그램을 통해 변수를 입력하면 DSP를 통해 신경망과 제어 알고리즘이 계산되어 출력되어지며 인터페이스 카드를 통해 모터 드라이버를 구동한다. 각각의 모

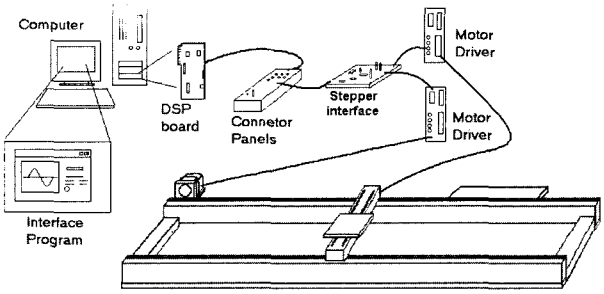


그림 2. 전체 시스템 실험 set-up.
Fig. 2. Experimental set-up.

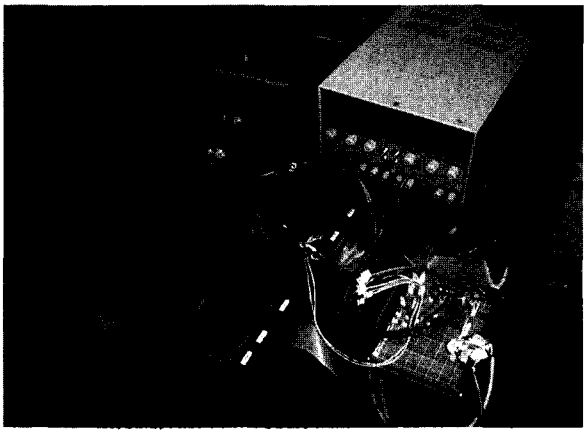


그림 3. DSP와 모터 드라이버와의 인터페이스.
Fig. 3. Interface between DSP and motor driver.

터 드라이버는 x-y 테이블의 각 축을 구동하게 되며 타이밍 벨트에 의해 구동축에 전달된다. 따라서 벨트의 장력이나 샤프트의 불균형에 의한 진동은 시스템의 동적 불확실성으로 간주된다. 각 DC 모터에 달려있는 엔코더를 통해 회전수를 알아내어 귀환하여 DSP의 제어 알고리즘에서 오차를 계산하여 사용하게 되므로 폐루프를 이루게 된다. DSP를 제어하기 위한 상용 소프트웨어를 사용하여 전체적인 샘플링 시간은 대략 180Hz 정도가 된다.

그림 3은 PC에 탑재되어 있는 DSP 보드와의 인터페이스를 나타내며 인터페이스를 통해 모터 드라이버를 구동시키는 실제 시스템을 보여준다. DSP로부터의 PWM 신호는 open collector 방식이므로 pull-up 저항을 통해 처리하였으며 신호의 전류를 증폭하기 위해 인터페이스 회로를 구성하였다.

2. 힘추종 실험환경

아래 그림 4는 x-y 테이블 위의 end-effector에 힘센서가 장착되어 있는 모습을 나타낸다. X-y 테이블은 2자유도를 가지므로 평면에서 수직한 힘만 제어하고자 한다. 로봇의 힘센서 끝의 end-effector는 ball-bearing이 장착되어 있어 마찰력을 최소화하였다. 대상의 강성도를 모른다는 가정 하에 로봇의 힘추종을 실험하기 위해 대상은 나무 합판과 철판, 두 가지를 선택하였다. 대상의 위치 또한 모른다 가정하였으므로 대상의 모양은 볼록한 형태로 하였다. 특히, 나무합판은 flexible하여 힘이 적용되면 합판이 자유자재로 움직이므로 대상의 위치가 시변하는 좋은 대상이 된다. 철판의 경우는 매우 단단하여 움직임이 거의 없다.

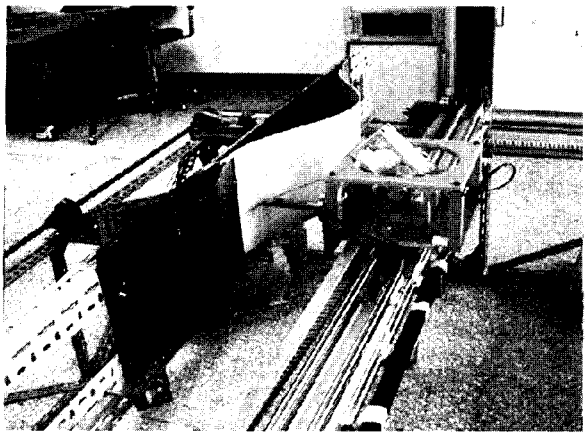


그림 4. 힘추종 실험 환경.
Fig. 4. Force tracking environment.

VII. 실험 결과

1. 접촉 힘제어의 경우

1.1 로봇이 나무판과 접촉하는 경우

먼저 로봇이 나무판에 5N의 힘을 가하는 경우를 보자. 나무판을 누르되 움직이지 않는 경우이다. 그림 5에서 로봇이 자유공간으로부터 움직여 나무와 접촉이 발생할 때 오버슈트가 나타나는 것을 볼 수 있다. 그림 5는 5N의 힘이 잘 조절되는 것을 나타내고 있다. 30초 정도에 힘이 다소 바뀌는 것은 나무판이 유연하므로 위치의 변형에서 오는 영향이라 할 수 있다. 두껍게 나타나는 것은 힘센서로부터 잡음을 나타낸다.

1.2 로봇이 철판과 접촉하는 경우

이번에는 로봇이 철판에 힘을 가하는 경우를 살펴보았다. 그림 6은 철판에 40N의 힘을 가했을 때의 결과이다. 접촉이 발생한 뒤에 오버슈트가 생기고 정착하는데 10초 정도 걸렸다. 이는 로봇의 end-effector 자체도 아주 rigid하지 않아 반발되는 힘에 의해 변형이 생기기 때문이다. 이 또한 신경회로망이 보상하여 정확한 힘을 조절하는 것을 그림 6에서 볼 수 있다. 다양한 실험으로 철판에 5N, 10N 등의 작은 힘을 적용하였지만 로봇이 대상과 계속적으로 접촉 비접촉을 반복하면서 불안정한 현상을 보였다. 이는 강성도가 큰 대상일수록 작은 힘을 제어하는 것이 어렵다는 것을 나타낸다.

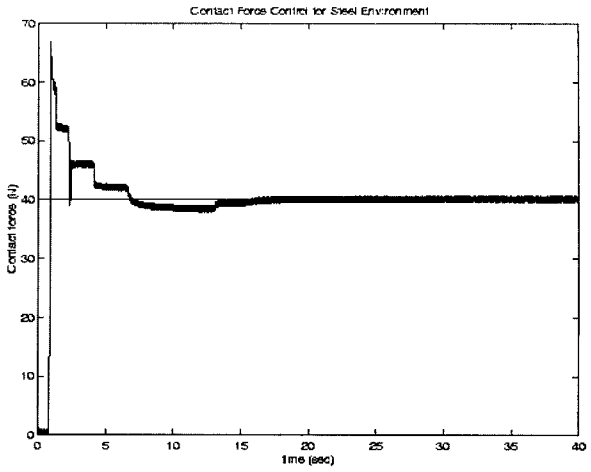


그림 6. 철판의 접촉힘 제어.
Fig. 6. Contact force control for steel.

2. 힘추종의 경우

2.1 로봇이 나무판위로 움직이는 경우

두 번째 실험으로 로봇이 나무판에 5N의 힘을 적용하면서 움직이도록 하였다. 이때 로봇은 나무판의 위치를 전혀 모른다고 가정하였다. 또한 나무판에 대한 강성도의 정보도 주어지지 않았다. 그림 7에 로봇이 힘을 추종하는 결과가 나타나 있다. 그림에서 보듯이 중간부분은 볼록한 나무판의 최고점을 통과하고 있는 것을 나타낸다. 이 점에서는 미끄러짐에 의한 비접촉이 발생하는 것을 볼 수 있다.

또한 나무판의 양끝에서도 나무판의 변형에 의해 힘오차가 크게 나타나는 것을 알 수 있다. 그림 8에서 보면 최고점의 높이는 대략 13cm 정도이다. 그림 8은 로봇이 추종한 나무판의 모양을 나타내고 그림 9는 신경회로망의 보상 출력을 나타낸다.

2.2 로봇이 철판위로 움직이는 경우

마찬가지로 로봇이 40N의 힘으로 철판 위를 움직이도록 하였다. 그림 10에 힘추종의 결과가 잘 나타나 있는데 나무판의 경우와는 달리, 가운데 부분에서 비접촉 현상이 발생하

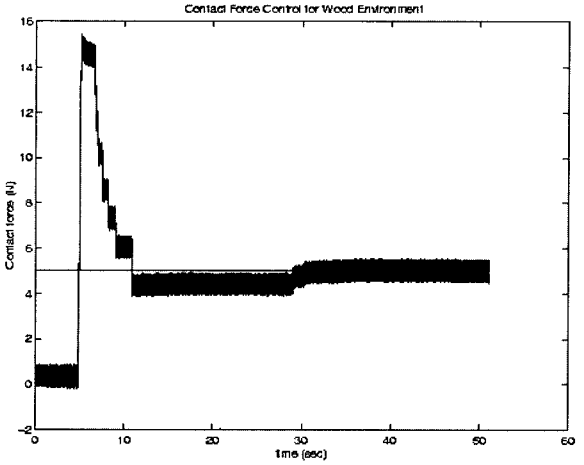


그림 5. 나무판의 접촉힘 제어.
Fig. 5. Contact force control for wood.

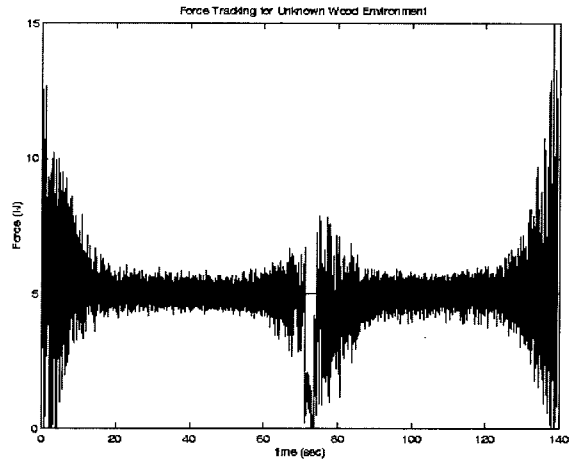


그림 7. 나무판의 힘추종 제어.
Fig. 7. Force tracking control for wood.

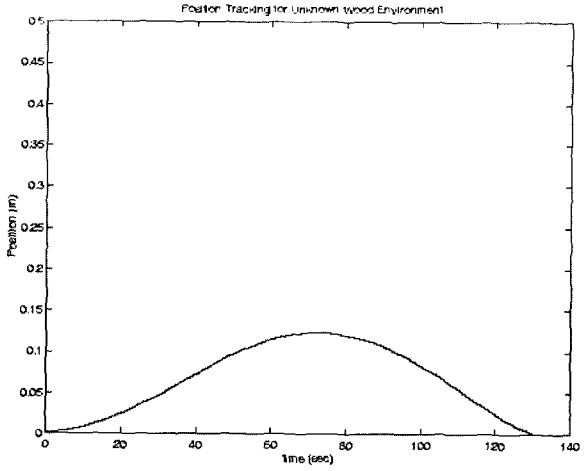


그림 8. 나무판의 위치 추종 제어.
Fig. 8. Position tracking control for wood.

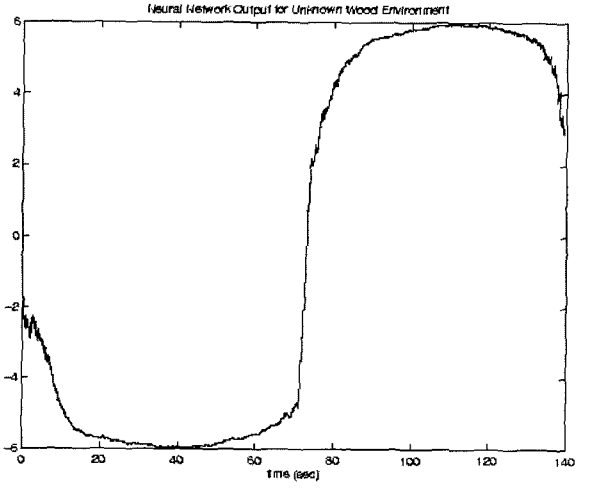


그림 9. 신경회로망 보상 출력.
Fig. 9. Neural network compensating signal.

지 않고 힘제어가 잘되고 있음을 알 수 있다. 이는 철판이 단단하기 때문에 모양에 변형이 발생하지 않기 때문이다. 그

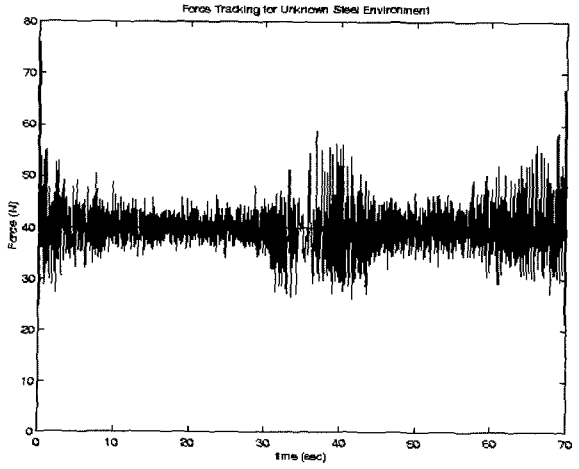


그림 10. 철판의 힘추종 제어.
Fig. 10. Force tracking control for steel.

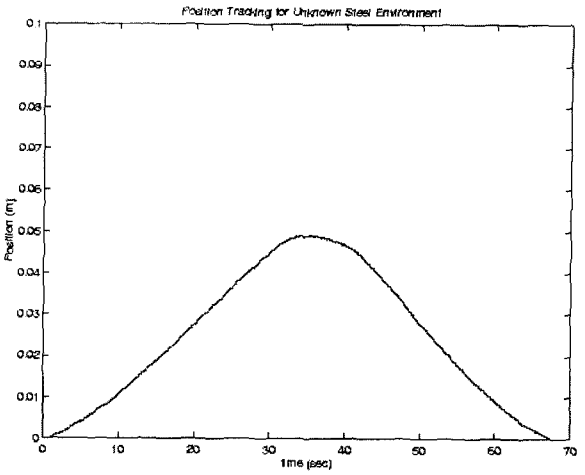


그림 11. 철판의 위치 추종 제어.
Fig. 11. Position tracking control for steel.

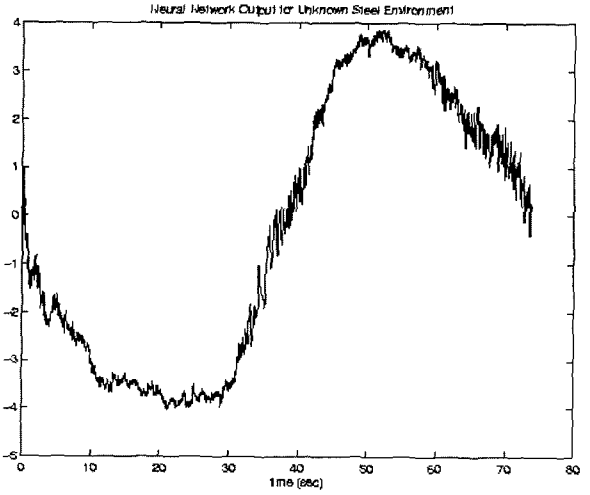


그림 12. 신경회로망 보상 출력.
Fig. 12. Neural network compensating signal.

림 11은 로봇이 추종한 철판의 모양을 나타내는 위치추종 결과인데 최고점의 높이는 대략 5cm인 것으로 실제와 같음을 알 수 있다. 그림 12는 불확실성을 보상하는 신경회로망의 보상신호를 나타낸다. 철판일 경우의 보상신호는 나무판일 경우의 보상 신호보다 더 진동이 있는 것을 알 수 있다.

VIII. 결론

본 논문에서는 신경회로망을 이용하여 힘제어시 발생하는 불확실성을 모두 보상하도록 하는 알고리즘을 제안하고 실험으로 검증하였다. 하나의 신경회로망으로 로봇의 동적 불확실성을 물론, 모르는 대상에서 오는 불확실성, 대상의 부정확한 위치 및 강성도까지 실시간으로 보상하였다. 나무합판의 경우 매우 유연하여 작은 힘도 제어가 가능하였으나 합판이 쉽게 움직이므로 최고점에서의 힘제어가 어려웠다. 철판의 경우에는 단단하여 작은 힘을 제어하기 어려웠으나, 최고점에서도 훌륭한 힘추종 결과를 보여주었다.

Acknowledgement

이 논문의 실험 설치를 도와준 석사과정 전풍우군과 지능 시

스템 및 감성공학 실험실 학생들에게 감사의 뜻을 전합니다.

참고문헌

[1] N. Hogan, "Impedance control : An approach to manipulator, part i, ii, iii," ASME *Journal of Dynamics Systems, Measurements, and Control*, vol. 3, pp. 1-24, 1985.

[2] M. Raibert and J. Craig, "Hybrid position/force control of manipulators," ASME *Journal. of Dynamic Systems, Measurements, and Control*, vol. 102, pp. 126-133, 1981.

[3] 정슬, "로봇의 디버링 작업이나 표면 광택 작업을 위한 새로운 힘제어 기술 개발," 제어 · 자동화 · 시스템 공학 논문지, pp. 786-795, 제6권, 9호, 2000년.

[4] 이보희, 이택승, 김진걸, "적응 학습률을 이용한 신경회로망의 학습 성능 개선 및 로봇 제어," 제어 · 자동화 · 시스템 공학 논문지, pp. 363-372, 제3권, 4호, 1997.

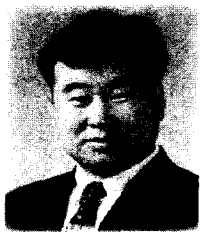
[5] 최부귀, 이형기, 박양수, "신경회로망을 이용한 유연한 관절의 선단위치 tracking 제어기에 관한 실험적 평가," 제어 · 자동화 · 시스템 공학 논문지, pp. 738-746, 제4권 6호, 1998

[6] 주진화, 이장명, "신경망을 이용한 이동 로봇의 실시간 고속 정밀 제어," 제어 · 자동화 · 시스템 공학 논문지, pp. 95-104, 제5권, 1호, 1999.

[7] 이민호, 최형식, "해저작업 로봇 매니퓰레이터를 위한 신경회로망을 이용한 슬라이딩 모드 제어기," 제어 자동화 시스템 공학 논문지, pp. 305-312, 제6권, 4호, 2000.

[8] 정슬, 임선빈 "신경회로망을 이용한 비선형 시스템 제어의 실험적 연구," 제어 · 자동화 · 시스템 공학 논문지, pp. 918-926, 제7권, 11호, 2001.

[9] S. Jung and T. C. Hsia, "A new neural network technique for robot manipulators," *Robotica*, vol. 13, pp. 477-484, 1995.



정 슬

1964년 9월 11일생. 1988년 미국 웨인주립대 전기 및 컴퓨터 공학과 졸업. 1991 미국 캘리포니아대 석사. 동대학 박사(1996). 1997년~현재 충남대학교 메카트로닉스공학과 조교수. 관심분야는 지능 제어 및 감성 공학, 로봇공학, 로봇 제작.

임 선 빈

1975년 10월 11생. 1998년 충남대 메카트로닉스공학과 졸업. 2000년 충남대 메카트로닉스공학과 석사. 관심분야는 신경망 제어 및 DSP 응용.